

20 LECCIONES DE BASIC
PARA APRENDER CON EL SPECTRUM





JACKSON

- TV y monitor
- Pantalla y memoria de imagen
- Colores y atributos
- INK, PAPER, BORDER, BRIGHT FLASH, OVER, INVERSE
- Funciones de control de la impresión en pantalla
- TAB, AT
- Números aleatorios con RND y RANDOMIZE
- Uso avanzado de PRINT e INPUT
- Videoejercicios
- Videojuego n.º 6



Spectrum
16K/48K/PLUS

VIDEO BASIC
Una publicación de
INGELEK JACKSON
Director editor por INGELEK:
Antonio M. Ferrer
Director editor por JACKSON HISPANIA:
Lorenzo Bertagnolio
Director de producción:
Vicente Robles
Autor: Softidea
Redacción software italiano:
Francesço Franceschini,
Stefano Cremonesi
Redacción software castellano:
Fernando López, Antonio Carvajal,
Alberto Caffarato, Pilar Manzanera
Diseño gráfico:
Studio Nuovaidea
Ilustraciones:
Cinzia Ferrari, Silvano Scolari,
Equipo Galata
Ediciones INGELEK, S. A.
Dirección, redacción y administración,
números atrasados y suscripciones:
Avda. Alfonso XIII, 141
28016 Madrid. Tel. 2505820
Fotocomposición: Espacio y Punto, S. A.
Imprime: Gráficas Reunidas, S. A.

ISBN del tomo 2: 84-85831-17-9
ISBN del fascículo: 84-85831-11-X
ISBN de la obra completa: 84-85831-10-1
Depósito Legal: M-15076-1985
Plan general de la obra:
20 fascículos y 20 casetes, de aparición quincenal, coleccionables en 5 estuches.
Distribución en España:
COEDIS, S. A.
Valencia, 245. 08007 Barcelona.
INGELEK JACKSON garantiza la publicación de todos los fascículos y casetes que componen esta obra y el suministro de cualquier número atrasado o estuche mientras dure la publicación y hasta un año después de terminada.
El editor se reserva el derecho de modificar el precio de venta del fascículo, en el transcurso de la obra, si las circunstancias del mercado así lo exigen.
Julio, 1985.
Impreso en España.

INGELEK



JACKSON

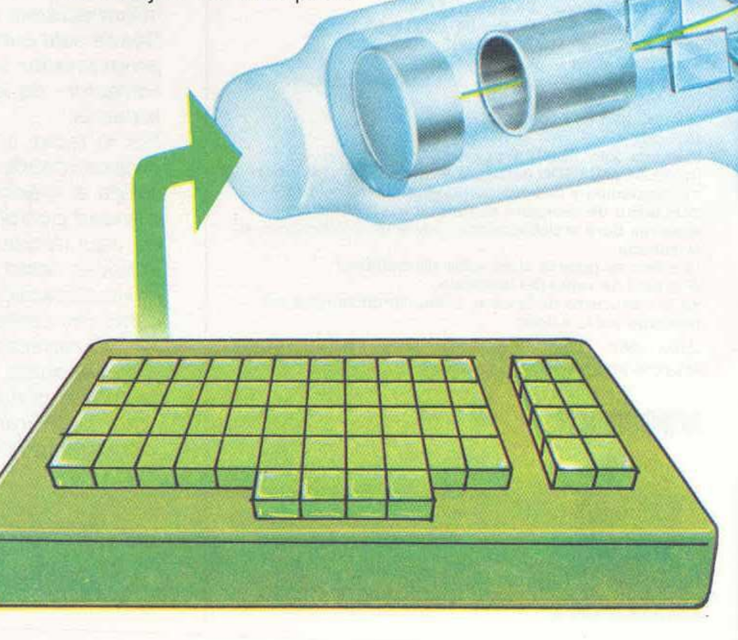
SUMARIO

Introducción

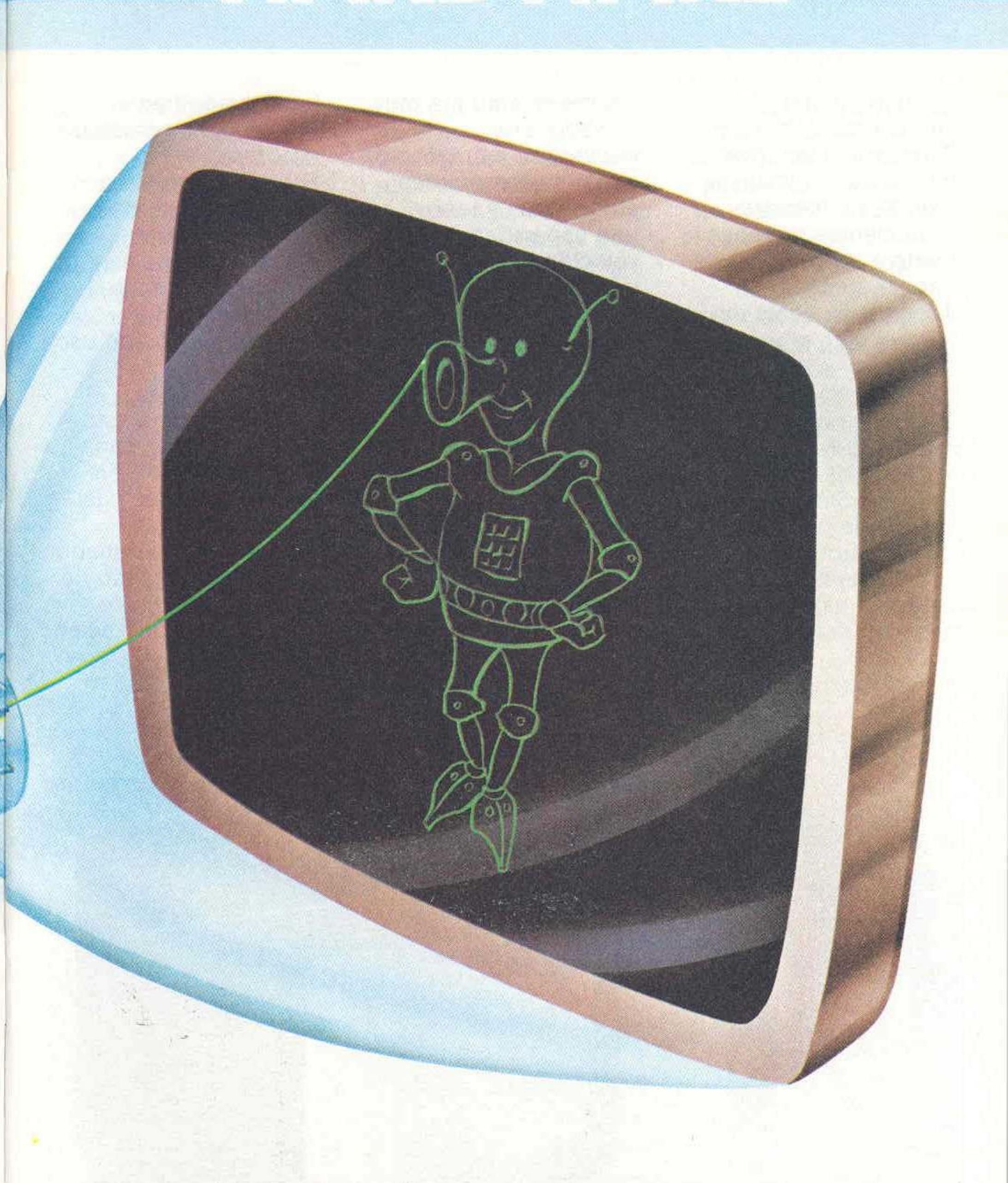
Entre otras cosas, tu ordenador es también una emisora de televisión. Es capaz de transmitir por cable las informaciones que ha elaborado. Desde este punto de vista, el programador se convierte en el «director» de la salida de los datos tratados.
Por lo tanto, él es el responsable de proporcionarle a las informaciones de salida el máximo relieve y la mayor claridad posibles.
De aquí deriva la importancia de conocer, tanto el hardware dedicado a la visualización de las imágenes, como las sentencias y funciones del BASIC capaces de manejar el formato y los atributos.
La mayor parte del éxito de tus futuros programas depende de tu familiaridad con estos elementos.

Televisores y monitores

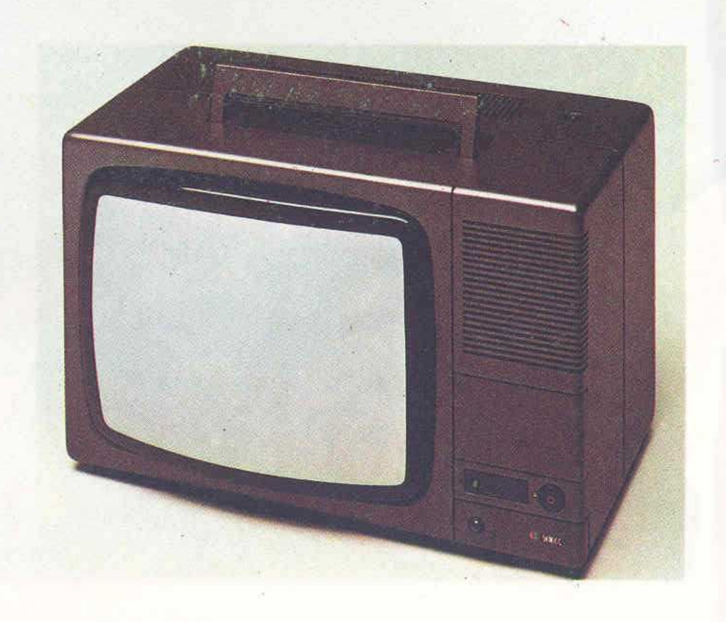
Televisor y monitor o, más generalmente, las unidades de visualización, constituyen el principal dispositivo de salida del ordenador.
Generalmente, es a estas a quienes se les asigna la tarea de visualizar todas las informaciones, los datos y los mensajes que constituyen la base fundamental de la relación entre el hombre y el ordenador.
Básicamente, las funciones desempeñadas por la unidad de visualización son tres:
1) Visualizar en pantalla la mayor parte de los caracteres tecleados (dicha función se llama eco).
2) Visualizar la salida de los programas, permitiendo así ahorrar tiempo y papel cuando no sea necesaria una copia permanente.
3) Enviar al usuario los mensajes del intérprete BASIC (por ejemplo: los mensajes de error).
El empleo de unidades de vídeo como dispositivos de visualización es bastante reciente; hasta hace aún pocos años las informaciones de salida de los ordenadores eran enviadas casi



HARDWARE

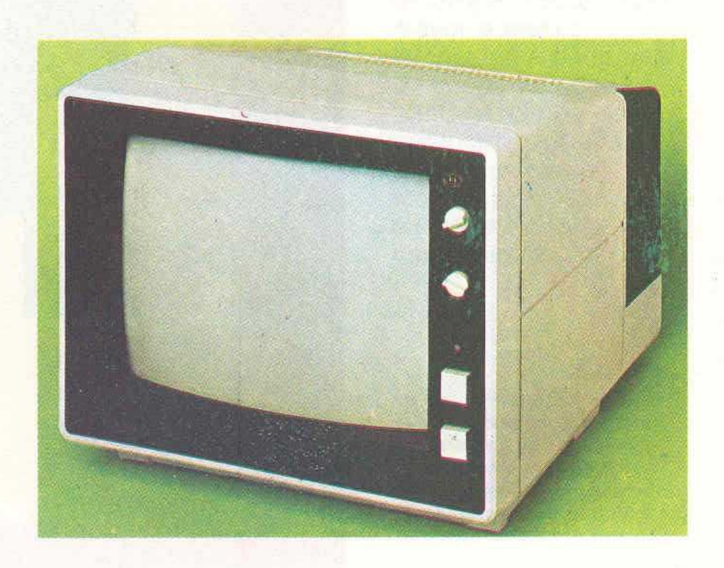


exclusivamente a impresoras y teletipos. Pero pronto se advirtió que estos dispositivos eran absolutamente insuficientes para hacer frente a un volumen de trabajo creciente. Además, su coste de mantenimiento era muy elevado (eran necesarias montañas de papel y permanentes atenciones) y tenían una capacidad y velocidad de visualización bastante limitadas con respecto a las exigencias de los usuarios.
Fue así, como se pensó en complementar las siempre necesarias impresoras con unidades de salida más adecuadas y flexibles en su uso que las empleadas hasta el momento.
La elección, como ya habrás adivinado, recayó en las pantallas de vídeo.
Estas respondían a todos los requisitos planteados: eran campactas, fiables, económicas (poco mantenimiento y mínimos gastos de uso), y rápidas.
Desde entonces, progresivamente, su empleo ha sido cada vez más intenso; en nuestros días es casi imposible lograr encontrar un ordenador que no esté dotado de su correspondiente unidad de visualización.



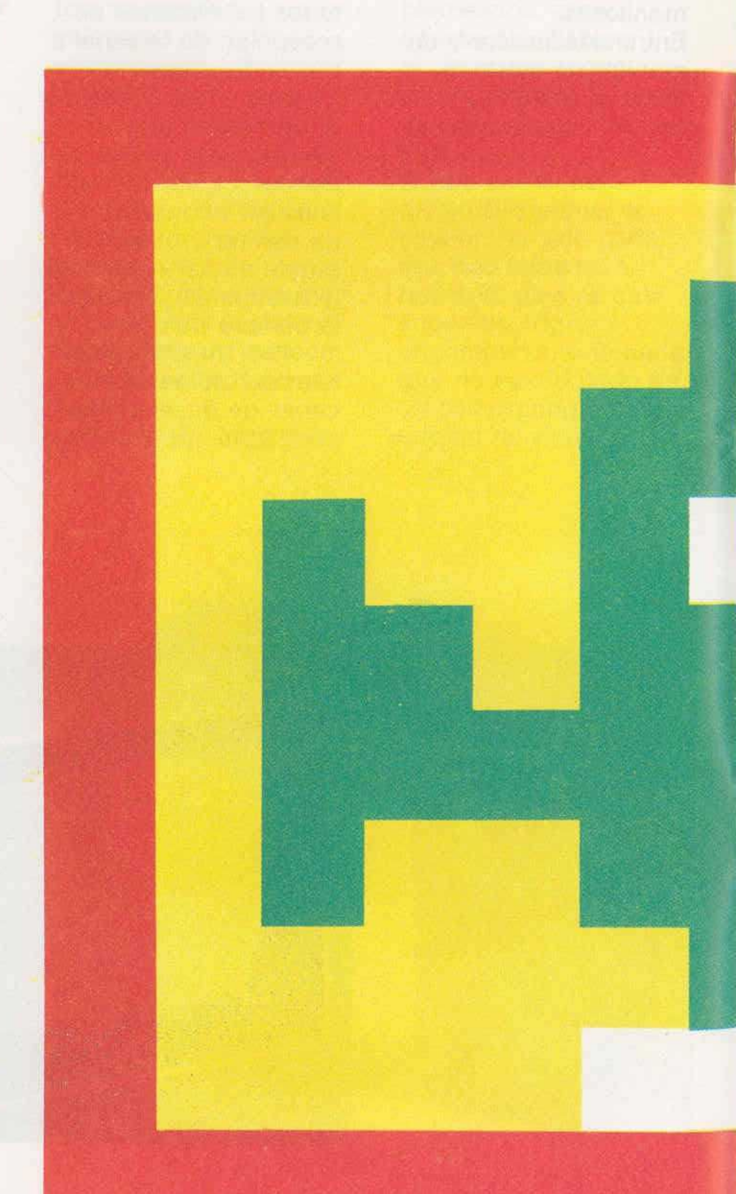
HARDWARE

En los personales modernos, las unidades de vídeo habitualmente empleadas son de dos tipos: televisores y monitores.
Entre un televisor y un monitor no existe ni física ni sustancialmente una gran diferencia. Un monitor no es más que una televisión de excelente calidad a la que se le han eliminado todos los circuitos de recepción de la señal a través de antena.
Naturalmente, la calidad de la imagen es superior a la que se puede obtener con un televisor normal. Para un uso no profesional puede no ser necesario (y hasta inútil) recurrir a la compra de un monitor, puesto que el televisor doméstico es capaz de desempeñar perfectamente el trabajo de visualización con un precio seguramente inferior.
La forma en que se producen los textos en la pantalla de tu Spectrum es bastante sencilla; existe una zona de la memoria RAM en la cual están depositados —bajo forma de códigos ASCII— todos los caracteres que pueden ser presentados en pantalla. Un circuito especial, en el interior de tu Spectrum, hace referencia a ella, realizando su «interface» con la unidad de vídeo.

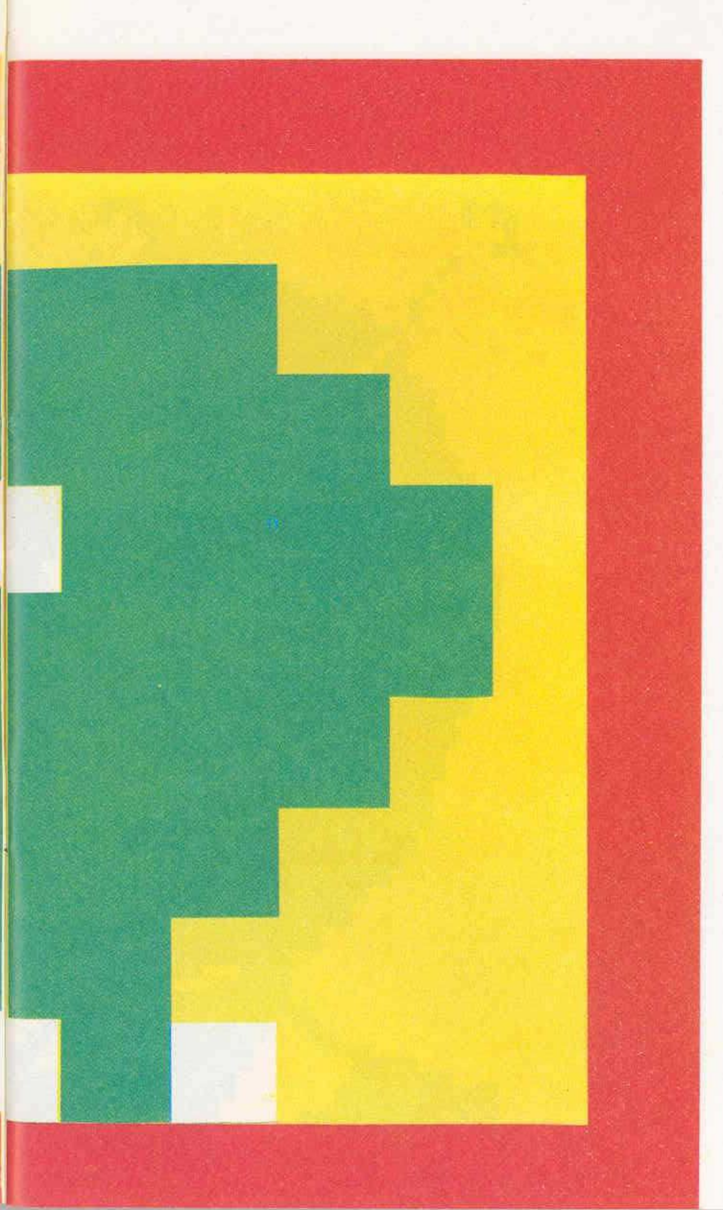


HARDWARE

Este circuito toma todos los caracteres presentes en las localizaciones de la memoria de pantalla y los envía al circuito del televisor, bajo forma de impulsos eléctricos compatibles con el sistema o «standard» de televisión.
El televisor (o el monitor) produce una imagen visible a partir de estas señales eléctricas.
La manera en que esto ocurre, constituye una de las aplicaciones más interesantes de la física electrónica y tiene como componente fundamental un dispositivo llamado tubo de rayos catódicos, que es un tubo al vacío, es decir, un contenedor de vidrio en cuyo interior se ha hecho el vacío. Dentro de este tubo se encuentra un «cañón electrónico» (basado en



un filamento calentado por la corriente que lo atraviesa, como en las bombillas) que produce un haz (o pincel) de electrones muy delgado.
Los electrones gozan de una propiedad singular: cuando chocan con determinadas sustancias fluorescentes –fósforo– provocan el que éstas generen una luminiscencia, cuya duración puede oscilar entre algunos milisegundos (milésimas de segundo) y algunos segundos, en función del tipo de fósforo y de la intensidad del haz electrónico.
La imagen se reconstruye a raíz precisamente de este hecho: el haz de electrones, «disparado» por el «cañón» y guiado por los oportunos campos eléctricos y magnéticos, desplazándose de derecha a izquierda y de arriba a abajo, aplica mayor o menor intensidad a cada uno de los puntos de una pantalla que ha sido recubierta por una finísima capa de fósforo, provocando una mayor o menor luminosidad.

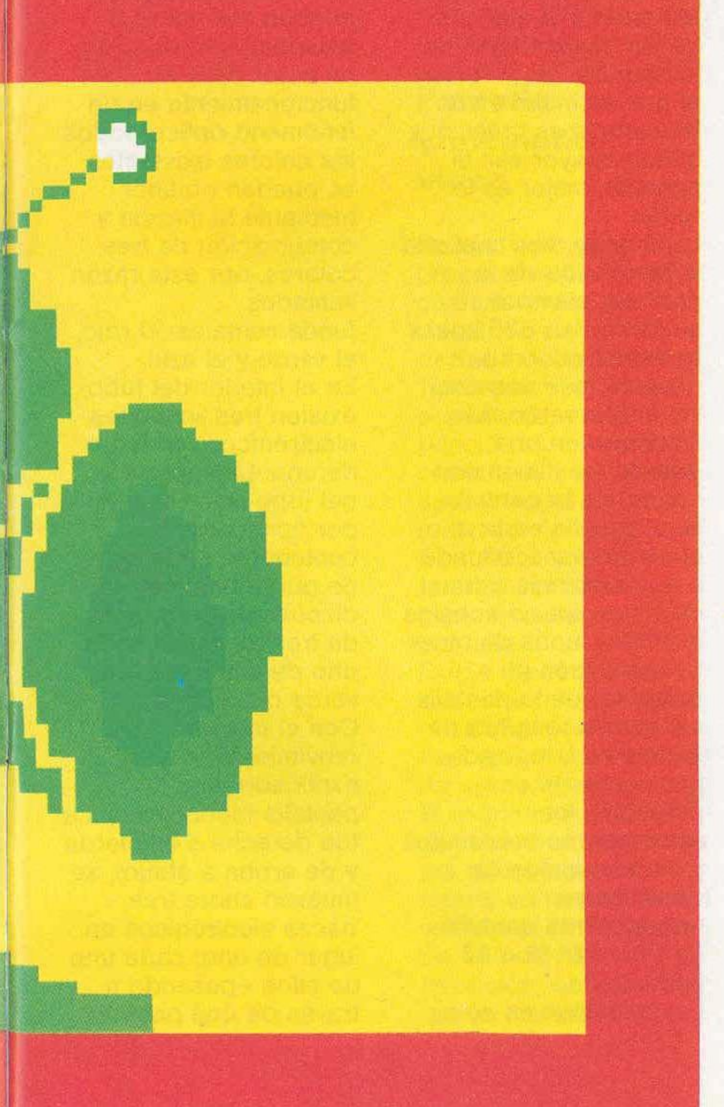


La baja resolución emplea únicamente los caracteres (normales o gráficos) existentes en el teclado. Las imágenes así obtenidas no resultan muy detalladas.

La imagen, por lo tanto, es reconstruida punto por punto por el pincel electrónico del tubo de rayos catódicos, de la misma forma en la que el ojo humano lee la página impresa de un periódico o de un libro. La velocidad con la que el pincel recorre la totalidad de la pantalla es tan extremadamente elevada, que no puede ser percibida siquiera mínimamente por el observador humano: el estándar europeo de televisión prevé que toda la pantalla, subdividida en 625 líneas horizontales (525 en el estándar americano) sea recorrida completamente cada cincuentavo de segundo.
El fenómeno de la persistencia de la imagen en la retina se encarga de



proporcionar una imagen completa y simultánea.
En el interior del televisor (o del monitor) existen, como ya se ha indicado, circuitos previstos para guiar los movimientos del haz de electrones tanto en sentido horizontal como vertical.
Para que la imagen sea visible es necesario que estos dispositivos trabajen simultáneamente.
En la señal de vídeo están compredidas, además de las informaciones referentes a la intensidad de cada punto de la pantalla, también señales especiales para coordinar el movimiento del pincel electrónico. Son las señales de sincronismo.
En las transmisiones de televisión, estas informaciones llegan desde la cámara; en cambio, para los ordenadores existe un circuito que, al igual que una cámara, «lee» la imagen que tiene que visualizar en las distintas posiciones de la memoria de pantalla.
El tipo de fósforo aplicado a la superficie de la pantalla determina el color del punto de



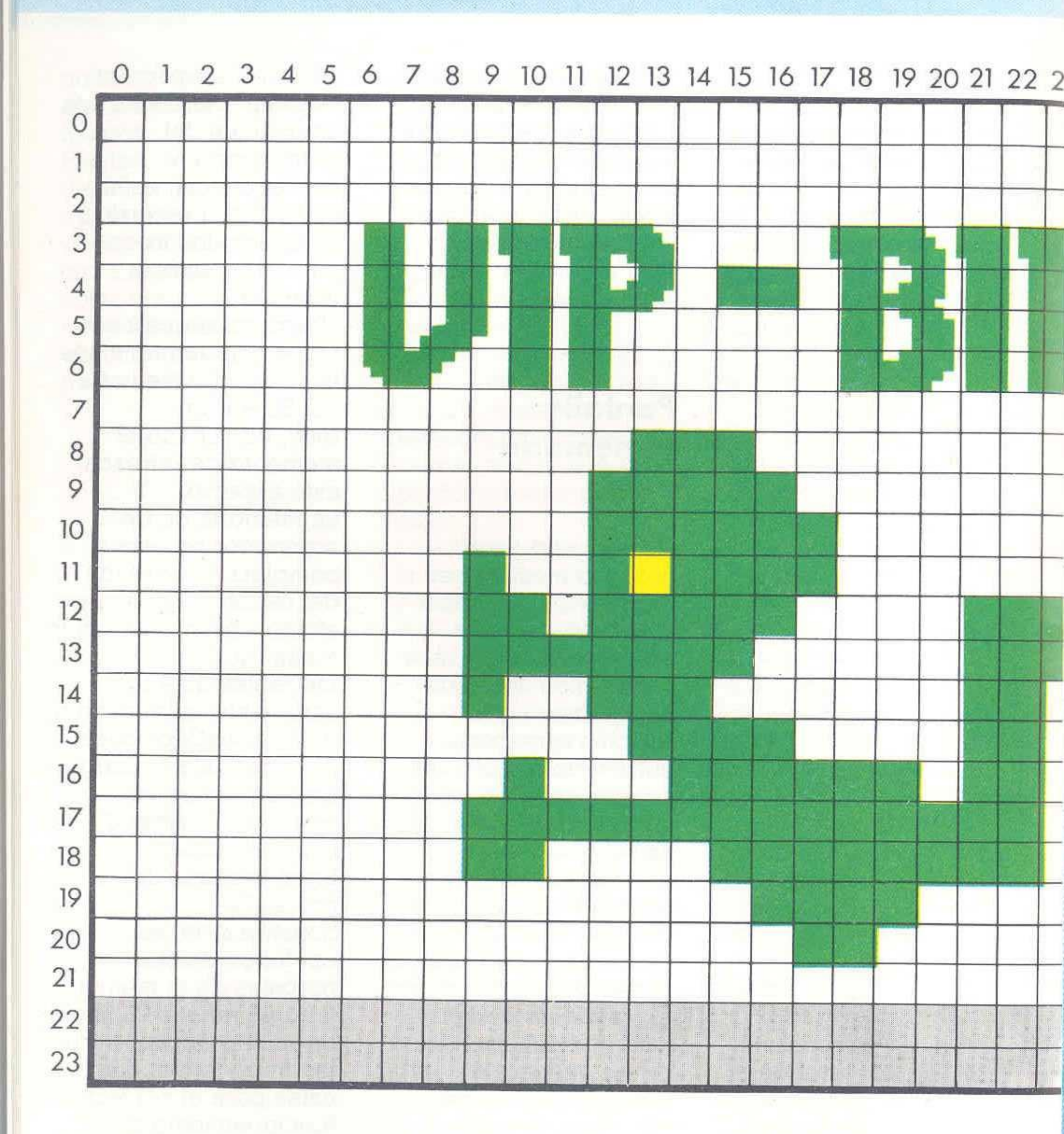
En alta resolución puedes direccionar cada punto (pixel) determinando o no su encendido.
Las imágenes así obtenidas resultan muy detalladas.

colisión entre el pincel electrónico y el propio fósforo.
Existen en el mercado distintos tipos de monitores: de fósforo verde, amarillo, ámbar, etc. La elección de un color u otro depende normalmente de una cuestión de gustos y preferencias personales, aunque últimamente lleguen continuos rumores (y desmentidos) sobre la mayor o menor fatiga visual producida por uno u otro tipo de fósforo.
En cambio, es muy importante la elección de las dimensiones de la pantalla. Un error en el que se incurre con frecuencia es creer que cuanto mayor sea la pantalla, mejor es la visión.
La imagen, sea cual sea la dimensión de la pantalla, siempre se subdivide en 625 líneas horizontales: en una pantalla más pequeña las líneas serán más finas que en una grande. La dimensión óptima de la pantalla, será aquélla en la que el ojo humano, situado a una distancia normal de visión, ya no consiga distinguir unas de otras.
La dimensión en pulgadas de la pantalla expresa la longitud de la diagonal, indicada precisamente en pulgadas: los ordenadores personales y los terminales de vídeo tienen normalmente pantallas de 9" o 12" (9 o 12 pulgadas).
Las pantallas en color merecen un tratamiento aparte.
Permaneciendo idénticos los principios de la televisión monocromática, es decir, de un solo color (habitualmente llamada «de blanco y negro» aunque sea verde o amarilla), una pantalla en color basa su funcionamiento en un fenómeno óptico: todos los colores existentes se pueden obtener mediante la mezcla y combinación de tres colores, por esta razón llamados fundamentales: el rojo, el verde y el azul.
En el interior del tubo, existen tres «cañones electrónicos» en lugar de uno. La superficie del tubo está recubierta por completo por centenares de millares de puntos de fósforo, dispuestos en grupos de tres, capaces cada uno de emitir luz roja, verde o azul.
Con el mismo movimiento, ya explicado para la pantalla monocromática (de derecha a izquierda y de arriba a abajo), se mueven ahora tres haces electrónicos en lugar de uno; cada uno de ellos —pasando a través de una pantalla

metálica adecuadamente perforada– es capaz de golpear únicamente el fósforo de un determinado color. Sobre la superficie de la pantalla aparecen por lo tanto tres imágenes simultáneas, que son, respectivamente, de color rojo, verde y azul, y que, combinándose entre sí proporcionan al ojo humano una sola imagen en colores. Esta técnica se llama síntesis aditiva.

Pantalla y memoria video

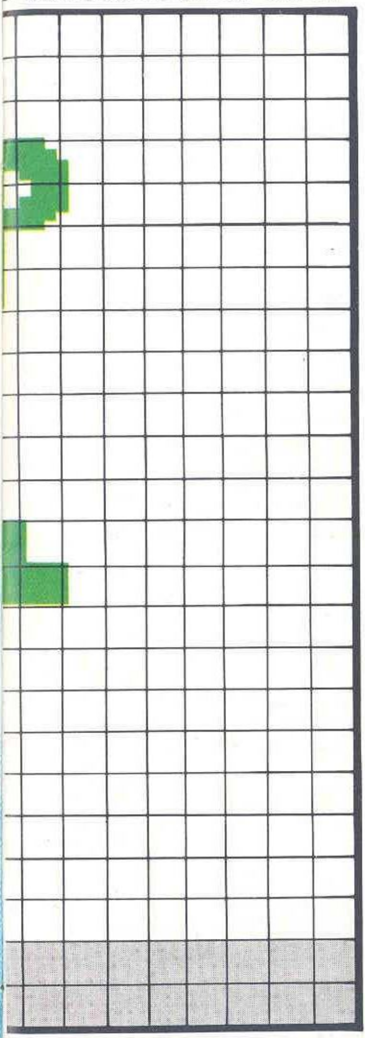
Hemos visto ya el proceso mediante el cual las imágenes se visualizan sobre la pantalla de tu televisor. Trataremos ahora de profundizar en la relación que existe entre todo aquello que se encuentra en la memoria de tu Spectrum, y lo que puedes ver representado en pantalla.
Existe de hecho una estrecha relación entre el contenido de la memoria y la formación de las imágenes.
El ordenador, para generar una imagen en pantalla, debe producir (como ya hemos dicho) una señal similar a la de una cámara de televisión, de tal forma que el monitor (o el televisor) no perciba ninguna diferencia. Los fabricantes del ordenador, en consecuencia, están obligados a recurrir a determinados trucos para «engañar» al televisor.
¿Recuerdas que hace algún tiempo hablamos del mapa de memoria del Spectrum?
Bien, ha llegado el momento de refrescar este aspecto.
La memoria de un ordenador no está a completa disposición del usuario: existen, en efecto, algunas zonas (o áreas) que no son utilizables directa y libremente para insertar datos e instrucciones, pero que, sin embargo, ejercen tareas que podríamos llamar de ayuda y sostén.
En el proceso de proyecto y construcción, se confiaron a estas parcelas de la memoria labores no exactamente de «elaboración», pero tan importantes como éstas para el correcto funcionamiento del aparato. Así, algunas localizaciones se han destinado a contener los programas, otras el intérprete BASIC, e incluso otras –y éstas



La pantalla del Spectrum está dividida en dos partes: la primera, de 22 líneas por 32 columnas, disponible para el BASIC, la segunda, de 2 líneas por 32 columnas, reservada para el sistema operativo, los INPUT y los mensajes.

son las que nos interesan ahora— los caracteres de visualización en pantalla.





El área de la memoria de la que nos vamos a ocupar ahora lleva el nombre de memoria de pantalla. Su objeto es contener todas las informaciones que son necesarias para construir una imagen sobre la pantalla de un televisor.
¿Crees qué esta memoria será RAM o ROM? No deberías tener demasiados problemas para adivinarlo.
Forzosamente, si tenemos en cuenta que lo que queremos es modificar el contenido de la pantalla, la memoria vídeo debe pertenecer a la zona de memoria RAM. En ella deben tener lugar los cambios siempre que, por ejemplo, ejecutes una sentencia PRINT o INPUT, o en el caso de que (excepto en ocasiones particulares)teclees algo que haya de ser visualizado en pantalla.
Pero, por si sola, la memoria de pantalla no puede hacer mucho: es necesario que un circuito especial de video llamado de «refresh» (es decir, refresco) de la imagen, lea —exactamente igual que una cámara— toda la memoria, carácter por carácter.
A partir de ese momento, el circuito de vídeo, conociendo ya los códigos de los caracteres a visualizar, consulta a una memoria especial ROM (llamada también generador de caracteres) de la que toma la «descripción» gráfica del carácter mismo. Como resultado final, se obtiene una señal de vídeo que indica si cada punto de la pantalla debe permanecer encendido o apagado.
Todo lo que aparece en pantalla es representado bajo la forma de determinadas combinaciones de puntos luminosos, llamados pixel (abreviatura del inglés de picture element).
Cada punto, pudiendo estar únicamente apagado o encendido, se puede describir con un único bit.
Sin entrar demasiado en detalles técnicos, te será suficiente con saber que el circuito de refresco «toma» la combinación de bits que contiene la descripción de un carácter y la coloca en

la casilla correspondiente de la pantalla, encendiendo o apagando los puntitos como ya se ha indicado.
En total, los pixels luminosos de que dispones en tu Spectrum son 49.152.
A ellos les corresponde, en conjunto, una pantalla compuesta por: 24 líneas y 32 columnas, que forman en total 24 x 32 = 768 caracteres.

Los atributos y los colores.

La pantalla es idealmente, y también físicamente, divisible en 768 (24 líneas de 32 caracteres) posiciones donde los caracteres pueden ser impresos.
Cada uno de ellos está representado por un cuadrado de 8 x 8 puntos.
Cada uno de estos caracteres se puede representar además de distintas maneras, por ejemplo: fijo, parpadeante, o coloreado.
El color, la luminosidad y el parpadeo son, para un carácter visualizado en una determinada posición de la pantalla, los llamados atributos.
Así, cuando imprimes algo en la pantalla no haces más que modificar la combinación de alguno de los atributos anteriormente asignados a aquella posición.
Normalmente, es decir, cuando no indicas especificaciones concretas —por ejemplo, al respecto del color—, el único atributo que sufre modificaciones es aquél referente al encendido o apagado de los pixels que componen el carácter.
Todos los demás atributos permanecen inalterados.
Pero existe también la posibilidad de variar a placer, mediante las instrucciones oportunas, todos los atributos.
En primer lugar, veamos el color.
He aquí la lista de colores disponibles en tu Spectrum, en el mismo orden en el que están indicados sobre las teclas numéricas:

---

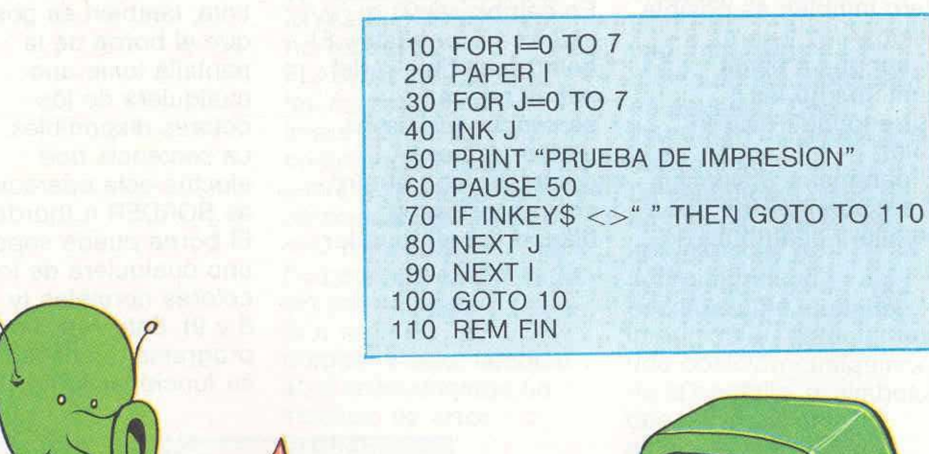
0 – negro
1 – azul
2 – rojo
3 – púrpura o magenta
4 – verde
5 – azul claro o cyan
6 – amarillo
7 – blanco

En un televisor en blanco y negro los números corresponden a una escala de grises, en el orden del más oscuro al más claro.
Cada carácter está asociado a dos colores: el color de tinta, o color de impresión, y el color del papel, o del fondo.
El color inicial de la tinta, en el momento del

encendido, es el negro, y el del papel el blanco. Con las instrucciones PAPER n (papel) e INK n (tinta) es posible alterar los colores.

Para ver su efecto ejecuta el siguiente programa (para terminarlo será suficiente con pulsar una tecla cualquiera):

```
 10 FOR I=0 TO 7
 20 PAPER I
 30 FOR J=0 TO 7
 40 INK J
 50 PRINT "PRUEBA DE IMPRESION"
 60 PAUSE 50
 70 IF INKEY$ <>" " THEN GOTO TO 110
 80 NEXT J
 90 NEXT I
100 GOTO 10
110 REM FIN
```



Por lo tanto, los paréntesis de PAPER e INK (n) pueden tomar todos los valores comprendidos entre 0 y 7.
Pero también es posible asignar los valores 8 y 9, aunque a estos números no les corresponda ningún color.
El parámetro 8 significa «transparente», es decir, no altera el atributo de la posición cuando se imprime el carácter; y todo queda como hubiera sido anteriormente especificado.
En cambio, el 9 significa «contraste». El color de la tinta o del papel, según la sentencia que hayas usado, se hace contrastar con el otro, convirtiéndose en blanco sobre un color oscuro (negro, azul, rojo, magental) y negro sobre un color claro (verde, cyan, amarillo o blanco).
Además del papel y la tinta, también es posible que el borde de la pantalla tome uno cualquiera de los colores disponibles.
La sentencia que efectúa esta operación es BORDER n (borde).
El borde puede soportar uno cualquiera de los 8 colores normales (y no 8 y 9). Este sencillo programa te enseñará su funcionamiento:

```
10 FOR I = 0 TO 7
20 BORDER I
30 PAUSE 50
40 NEXT I
50 REM FIN
```

Las dos últimas sentencias que modifican los atributos son BRIGHT n (luminosidad) y FLASH n (parpadeo).
BRIGHT n modifica la luminosidad de los caracteres impresos a partir del momento en que se ejecuta la sentencia en adelante: n=0 para luminosidad normal, n=1 para luminosidad extra y n=8 para trasparencia.

Si n no fuera 0, 1 u 8, obtendremos el mensaje de que el número indicado no es un valor permitido: aparecerá en pantalla INVALID COLOUR (color no válido).
FLASH n establece si los caracteres impresos a continuación de la sentencia han de ser parpadeantes o fijos; n=0 para fijos, n=1 para parpadeantes, n=8 para ningún cambio.
Si n no fuera un número válido, también aquí obtendríamos un mensaje de error.
Existen otras sentencias, INVERSE, y OVER, que no modifican los atributos, pero que actúan sobre la matriz de puntos impresa en pantalla.

Usan únicamente los parámetros 0 y 1, al igual que FLASH y BRIGHT. Como ya te habrás imaginado si empleas la instrucción INVERSE 1, tu Spectrum imprimirá desde ese momento los caracteres en negativo. Con INVERSE 0 volverás a la normalidad.
La sentencia OVER 1, en cambio, activa un tipo especial de sobreimpresión.
Normalmente, cuando imprimes un carácter en una posición cualquiera de la pantalla, el símbolo que allí estuviera anteriormente representado es borrado y sustituido por el último. Con OVER 1, en cambio, el nuevo carácter es sumado, es decir, superpuesto al anterior. Esto te permite, por ejemplo, subrayar un carácter. Haz la prueba:

```
10 PRINT "A"; CHR$ (8);:REM imprimir el carácter cuyo código es 8
   equivale a volver atrás con el cursor.
20 PRINT "_": REM el símbolo _ se obtiene pulsando SYMBOL SHIFT
   y 0.
```

Después con

```
10 PRINT "A";CHR$(8);
20 PRINT OVER 1; "_"
```

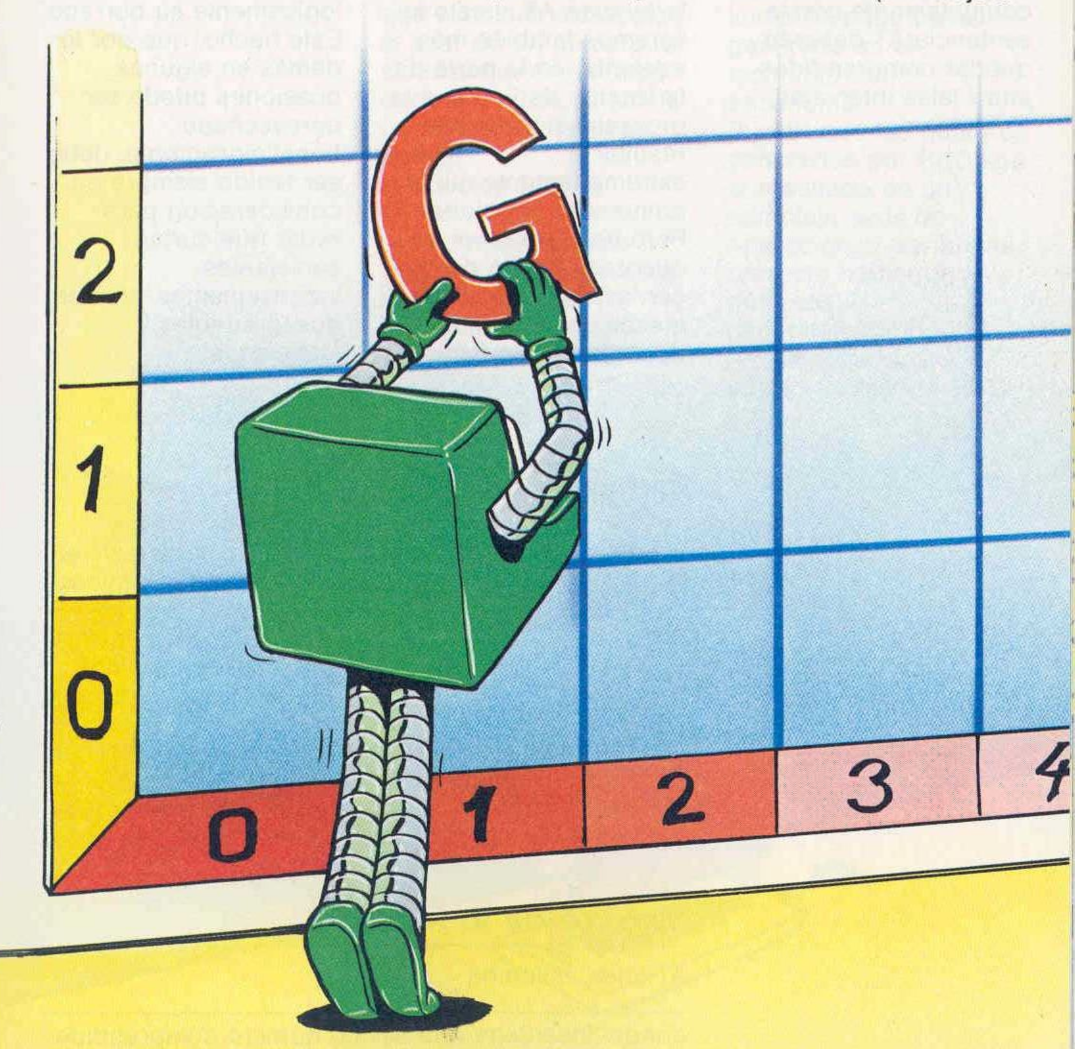
¿Has visto la diferencia?

Funciones de control de la impresión en pantalla

En el ámbito de las instrucciones de visualización previstas por el BASIC, revisten una especial importancia todas aquellas sentencias que controlan y modifican, al gusto del programador, la posición del cursor y, en consecuencia, la de los letreros en pantalla.
Hasta ahora hemos empleado la instrucción PRINT en numerosas ocasiones, usándola para visualizar todos los resultados, los mensajes y los letreros que hayan sido necesarios cada vez.
Pero lo que todavía nos falta es la capacidad de controlar completamente esta instrucción, lo que nos permitiría, por ejemplo, conseguir que los resultados de salida estuvieran colocados en una determinada posición de pantalla, o bien ordenados y encolumnados con formatos no impuestos necesariamente por el ordenador.
Dicho de otra manera (con una palabra muy empleada en el campo informático), deseamos saber formatear los textos en pantalla.
Es pues éste el objetivo de las instrucciones que ahora nos disponemos a aprender.
Y es importante subrayar que todas estas instrucciones no le indican al ordenador qué imprimir, sino únicamente DONDE imprimir.
¿Has entendido la diferencia?
Veámoslas ahora una a una, junto a algunos ejemplos aclaratorios.

AT

El objeto de la función AT es el de desplazar la posición de impresión (es decir, el punto de la pantalla donde será impreso el siguiente carácter) a la línea y columna especificadas. Ya hemos visto como la pantalla de tu Spectrum dedicada al BASIC está dividida en 22 líneas y 32 columnas.
Las líneas (de abajo a



arriba) han sido numeradas de 0 a 21, y las columnas (de izquierda a derecha) de 0 a 31.
La línea y la columna que especifiques conjuntamente con la sentencia AT deberán quedar comprendidas entre tales intervalos, para no recibir el mensaje de error INTEGER OUT OF RANGE (fuera de la pantalla).
Una última observación. Ya habrás intuido que la función AT, y esto lo veremos también más adelante, en la parte de la lección dedicada a la programación, puede resultar extremadamente útil en numerosas ocasiones. Pero hay que tener en cuenta el hecho de que con ella se alteran los mecanismos normales de escritura en pantalla; por lo que puede ocurrir que una sentencia AT, indicada sin pensarlo mucho, lleve a escribir ENCIMA de un texto ya existente en pantalla, provocando lógicamente su borrado. Este hecho, que por lo demás en algunas ocasiones puede ser aprovechado beneficiosamente, debe ser tenido siempre en consideración para evitar que surjan semejantes inconvenientes, siempre desagradables y fastidiosos.

Ejemplos

PRINT AT 10, 15; "GATITO"	Imprime la palabra «gatito» exactamente en el centro de la pantalla.
PRINT AT 21, 27; "GATO"	Imprime la palabra «gato» en la esquina inferior derecha de la pantalla.
PRINT AT 0,27; "GATO"	Esta vez el felino aparecerá en la esquina superior derecha.

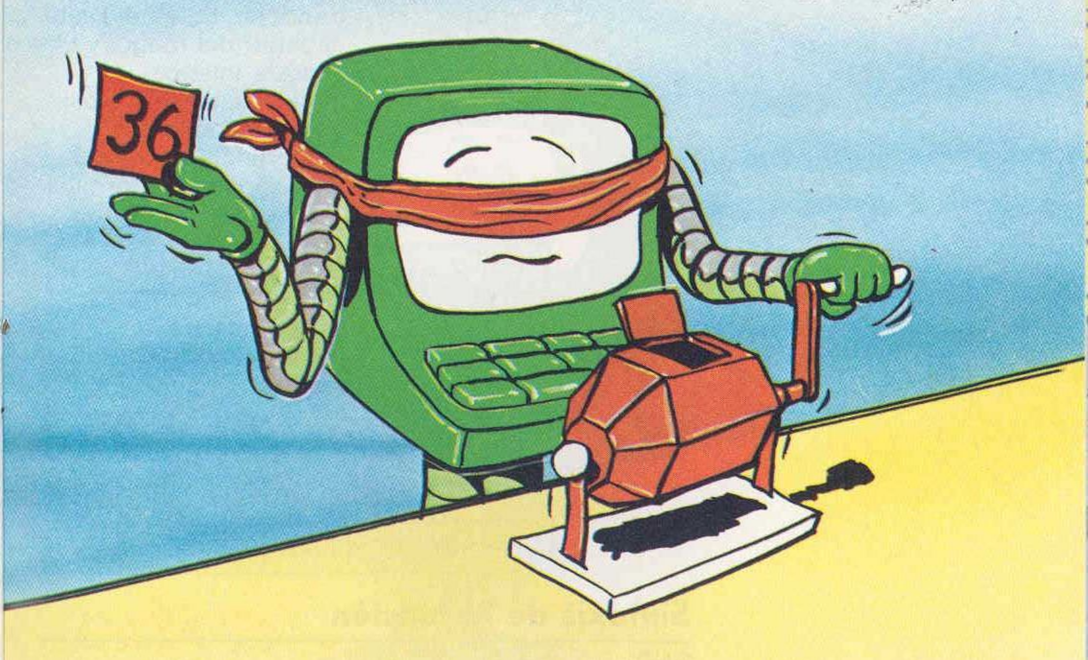
Sintaxis de la función

AT línea, columna

donde línea tiene que ser un número comprendido entre 0 y 21 y columna entre 0 y 31.

RND

No es raro encontrar casos en los que sea necesario disponer de números aleatorios dentro de un programa. Por ejemplo, podrías necesitar un algoritmo que simule la extracción al azar de un número, o el lanzamiento de unos dados, o quizá, la salida de las cartas de una baraja.
La función RND te resuelve excelentemente este tipo de problemas: crea números al azar (random en inglés) comprendidos entre 0 y 1 (0 < = R < 1).
Tu Spectrum produce una secuencia de números aleatorios realizando una serie de operaciones a partir de un número inicial llamado base, generado en el momento del encendido.
Puesto que los números generados por RND son el resultado de una compleja serie de operaciones, sería más correcto hablar de números pseudoaleatorios.
A diferencia de las demás funciones, RND



no necesita de argumento: por lo tanto, PRINT RND es una instrucción correcta y determina la impresión de un número aleatorio. Copia y prueba el siguiente programa para darte cuenta de su funcionamiento:

```
10 FOR C=1 TO 100
20 PRINT C; " "; RND,
30 NEXT C
```

Puesto que la gama de números comprendidos entre 0 y 1 no resulta adecuada para la mayor parte de las aplicaciones, emplea la siguiente fórmula para establecer por ti mismo el rango dentro del cual habrán de ser producidos los números aleatorios:

```
LET R = INT ((LS – LI + 1) * RND) + LI
```

donde R significa número aleatorio (random), LS es el límite superior del rango, y LI el límite inferior.
Para obtener números aleatorios enteros comprendidos entre 9 y 18, aplicando la fórmula obtendrás:

```
LET R = INT ((18 – 9 + 1) * RND) + 9
```

Para números comprendidos entre 0 y 10 será:

```
LET R = INT ((10 – 0 + 1) * RND) + 0
```

es decir:

```
LET R = INT (11 * RND)
```

Sintaxis de la función

RND

RANDOMIZE

La instrucción RANDOMIZE se emplea fundamentalmente para determinar desde que punto de la secuencia tiene que empezar a extraer números aleatorios la función RND.
Prueba ahora con:

```
10 RANDOMIZE 22
20 FOR C = 1 TO 10
30 PRINT RND
40 NEXT C
```

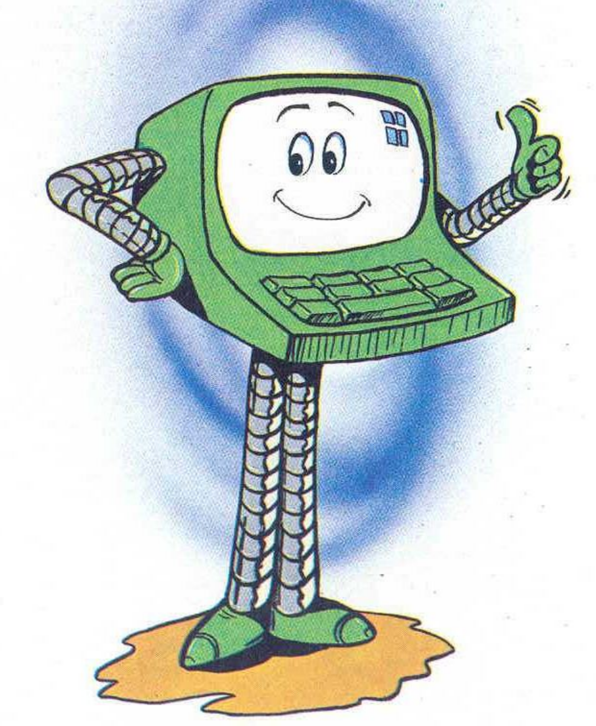
Anota los diez números aleatorios, después pon de nuevo en marcha el programa con RUN, y compáralos.
Modifica ahora en la línea 10 el número que sigue a RANDOMIZE, y vuelve a intentarlo...
¿Has visto la diferencia?
Cambia de nuevo la línea 10.
Escribe:

```
10 RANDOMIZE
```

o bien

```
10 RANDOMIZE 0
```

sin modificar las líneas restantes.
Pon nuevamente en marcha el programa varias veces, anotando



y comparando las distintas series de números: no encontrarás nunca una idéntica a la otra.
Resumiendo:

- Si deseas generar series de números aleatorios iguales entre ellas, coloca antes de la función RND una instrucción RANDOMIZE, seguida siempre por el mismo número.
- Si en cambio, deseas obtener secuencias de números aleatorios siempre distintas, sitúa antes de RND la instrucción RANDOMIZE 0, o simplemente RANDOMIZE.



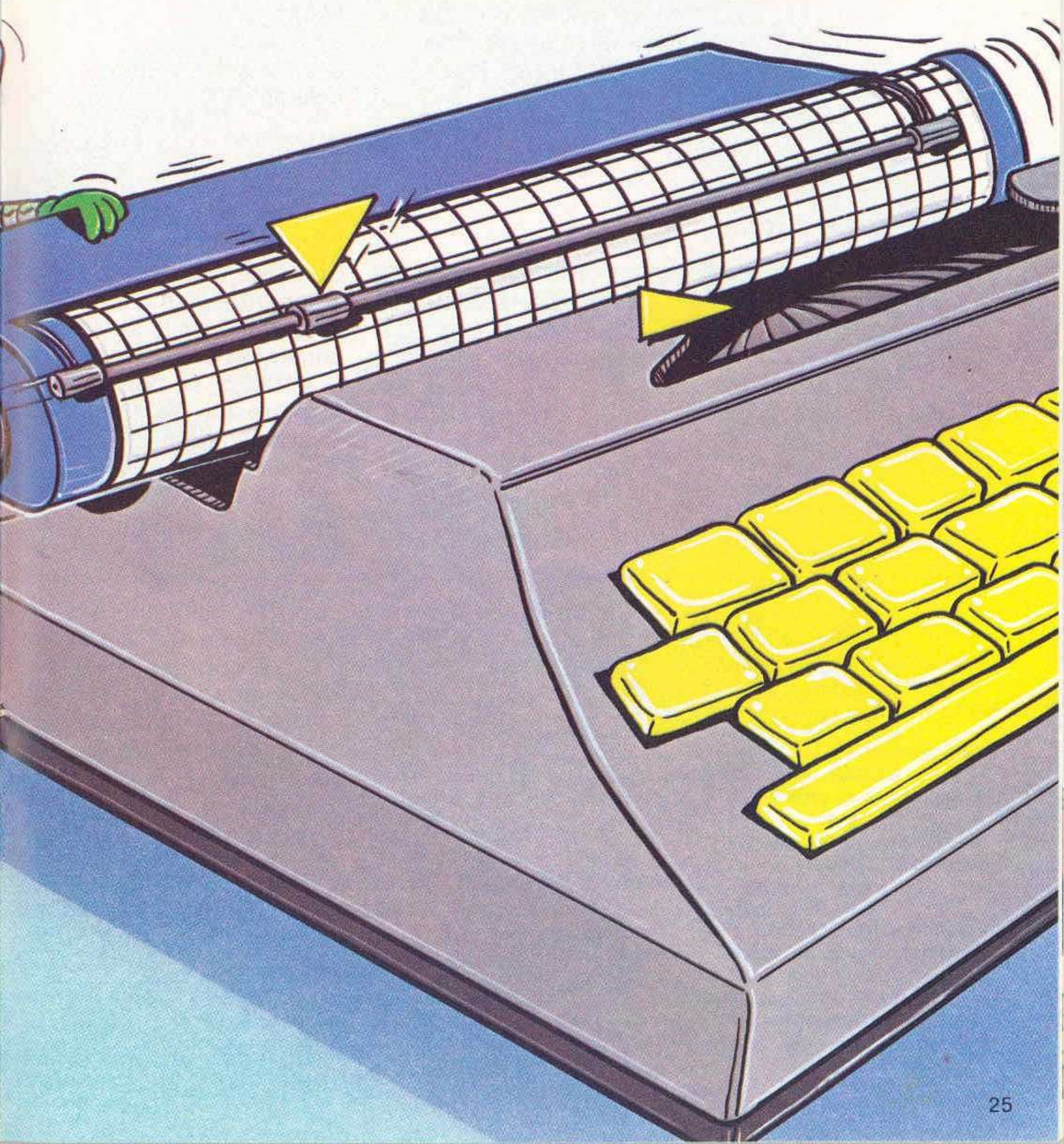
Sintaxis de la instrucción

RANDOMIZE [número]

TAB

Igual que AT, también la función TAB() se emplea únicamente en el interior de PRINT: opera como los tabuladores de una simple maquina de escribir.

La función TAB() se emplea normalmente para alinear las impresiones en columnas verticales, alineándolas en puntos preestablecidos.



Supón que tengas que visualizar algunos datos en pantalla, alineados en un orden determinado. Gracias a TAB() puedes evitar los aburridos y a veces complicados cálculos para encolumnar con exactitud todos los elementos.

TAB() permanece sobre la misma línea donde se encuentre el cursor, excepto cuando la posición especificada de impresión no sea anterior a la posición actual.
En este caso se desplazará a la línea siguiente.

Ejemplos:

```
PRINT NOMBRE$; TAB(13); APELLIDO$
```

Con esta instrucción e independientemente de la longitud de la cadena contenida en NOMBRE$, se obtendrá la impresión de dos variables con la siguiente disposición: la primera a partir de la columna 0, la segunda desde la 13.

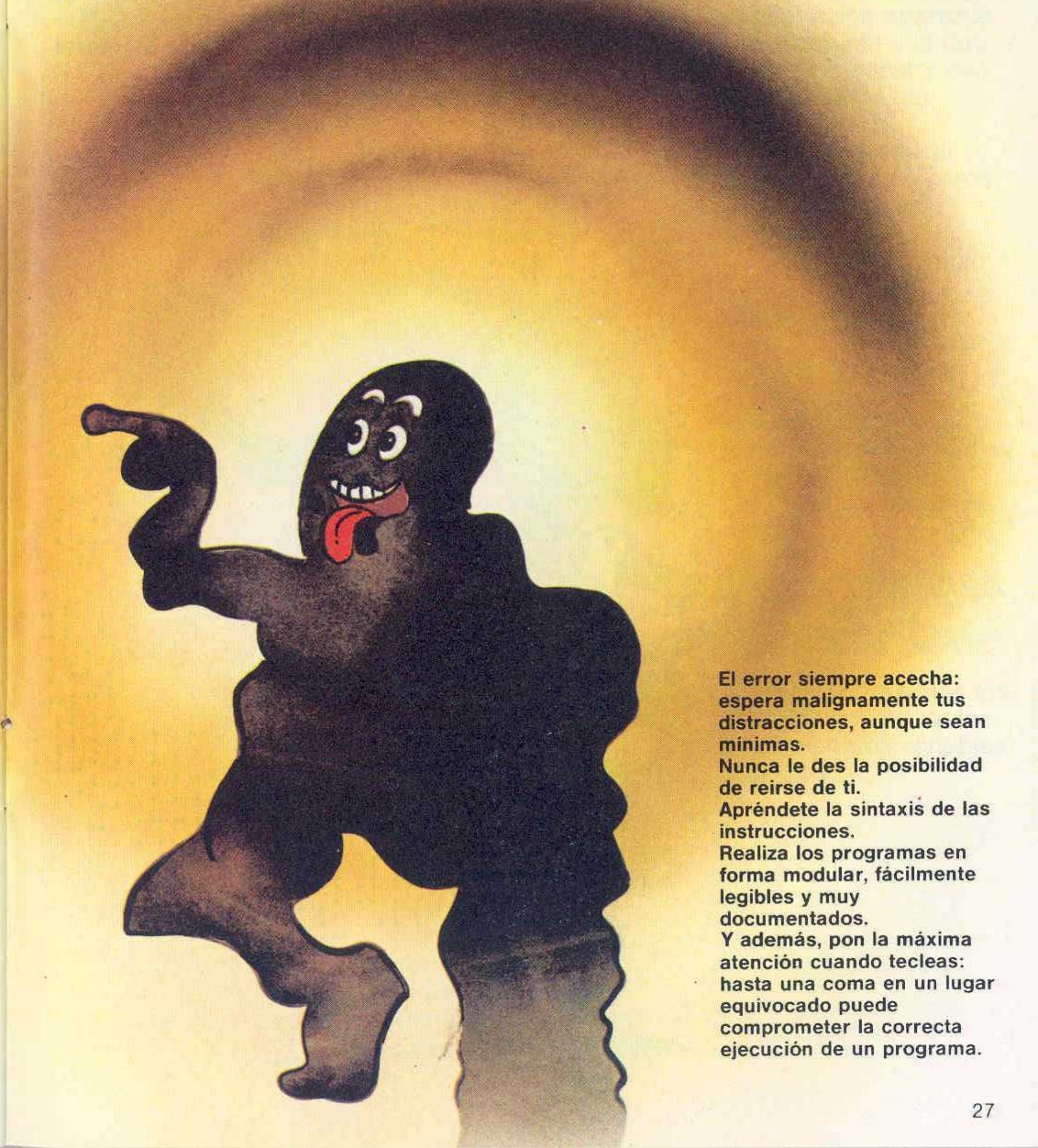
```
10 PRINT TAB(2); "NUMERO"; TAB(12);
   "CUADRADO"
20 FOR I=1 TO 15
30 PRINT TAB(4);I; TAB (15);I ↑ 2
40 NEXT I
```

Este programa muestra una sencilla aplicación de TAB: imprime encolumnados los primeros 15 números con sus respectivos cuadrados.

Sintaxis de la función

TAB (expresión)

LENGUAJE



El error siempre acecha: espera malignamente tus distracciones, aunque sean mínimas.
Nunca le des la posibilidad de reirse de ti.
Apréndete la sintaxis de las instrucciones.
Realiza los programas en forma modular, fácilmente legibles y muy documentados.
Y además, pon la máxima atención cuando tecleas: hasta una coma en un lugar equivocado puede comprometer la correcta ejecución de un programa.

Uso avanzado de PRINT e INPUT

Ahora aprenderás a usar ventajosamente las varias instrucciones de ENTRADA/SALIDA que hemos ido viendo. Hablaremos del uso avanzado de PRINT e INPUT.
La palabra «avanzado» permite intuir sus numerosas posibilidades y, de hecho, viene a indicar el conjunto de reglas, secretos y trucos que le permiten al programador obtener el control pleno y



absoluto de todo lo que entra o sale del ordenador.
Una primera y muy útil instrucción sería:

```
POKE 23692,255
```

esto evitará que la máquina, después de haber llenado la pantalla de letras o números escriba siempre SCROLL? en la línea inferior, parando en consecuencia la ejecución del programa.
Por ejemplo, intenta:

```
10 FOR X=65 TO 127
20 PRINT X; TAB 8; CHR$(X)
30 POKE 23692,255
40 NEXT X
50 IF INKEY$<>" " GO TO 10
```

Descubrirás que la visualización de los caracteres creados en el interior del bucle FOR por la sentencia CHR$ no sufrirá ninguna parada durante su ejecución.
La única manera de parar el programa (línea 50) será pulsando una tecla cualquiera.
Para restablecer el procedimiento normal de scroll teclea:

```
POKE 23692,1
```

Veamos ahora algunas cosas sobre INPUT. Esta instrucción es capaz de hacer algunas cosas más de las que has visto hasta ahora.
Supón que desees escribir un programa para aprender a hacer multiplicaciones mentalmente.
Deseas que como entrada se te pida una pareja de números cualquiera y que a continuación aparezca una pregunta a la que deberás contestar con el resultado de la multiplicación.
El problema será el siguiente: los números de entrada serán asignados a dos variables (por ejemplo A y B).
¿Cómo podremos asignar a una tercera variable (llamémosla C) el número proporcionado como resultado, mediante una sentencia INPUT?
Una línea de programa como:

```
INPUT "¿CUANTO DARA"; A; "*";B; "?"; C
```

consideraría a todos los elementos fuera de las comillas como variables a asignar.

Esta dificultad (¡tan larga de explicar!) se supera en cambio, rápida y fácilmente: bastará con encerrar ente paréntesis los nombres de las variables:

```
INPUT "¿CUANTO DARA"; (A); "*"; (B); "?"; C
```

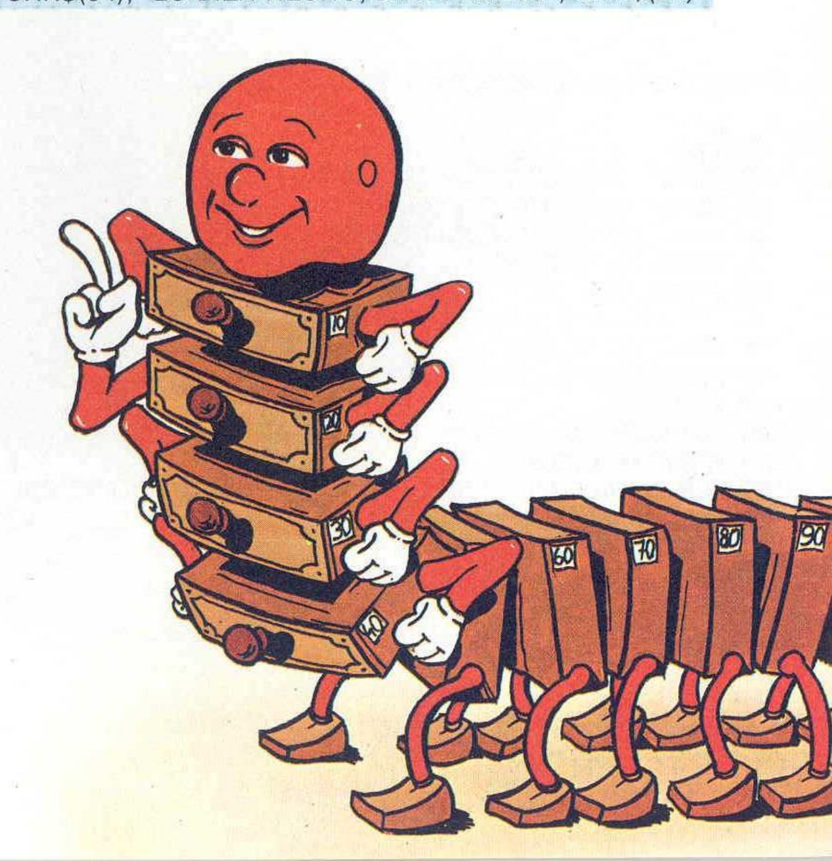
¡Ya está todo resuelto! Es bien sencillo y esta regla es general. Cualquier expresión que empiece con una letra, y que deba ser impresa como parte de un mensaje, debe ser encerrada entre paréntesis. Así, el programa completo podría ser de este estilo:

```
 10 LET LINEA=10
 20 CLS: INPUT AT 10,0; "¿CUAL ES EL PRIMER NUMERO?  "; A
 30 INPUT AT 12,0; "¿CUAL ES EL SEGUNDO NUMERO?  "; B
 40 CLS INPUT AT LINEA, 0; "¿CUANTO DA"; (A); "*"; (B); "?"; C
 50 IF C=A*B THEN PRINT AT LINEA+4,0; "¡¡BIEN!!": GO TO 100
 60 PRINT "¡¡ERROR!!"
 70 LET LINEA=LINEA+2
 80 IF LINEA <=14 THEN PRINT TAB 15; "VUELVELO A INTENTAR";
    PAUSE 100: GO TO 40
 90 PRINT AT 21,0; "EL RESULTADO DE"; A; "*"; B; "ES", A*
100 INPUT "¿QUIERES SEGUIR?  "; R$:IF R$="S" THEN RUN
110 REM FIN
```

El listado no requiere demasiados comentarios: la única cuestión de relieve aparece en la línea 80. Esta línea está constituida por varias instrucciones (debidamente separadas por los dos puntos), la primera de las cuales es una IF. Ten en cuenta que si la condición de IF fuera falsa (es decir, no

verificada), en aquella línea no se ejecutará ninguna de las demás instrucciones: el intérprete BASIC se salta toda la línea, llevando la ejecución a la instrucción siguiente (es decir, a la línea 90).
Otra cosa, esta vez tanto acerca de PRINT como de INPUT: ocurre a veces que se desea imprimir en pantalla alguna palabra encerrada entre comillas. Pero esto no será posible en condiciones normales. Las comillas (" ") en el BASIC se emplean como delimitadores: cada vez que el intérprete BASIC encuentre una, la entiende como carácter inicial o final de una constante de cadena. ¿Qué hacer entonces? La respuesta se llama código ASCII.
Será suficiente con emplear el código ASCII de las comillas (que es el 34) para «saltarse» al intérprete BASIC.
Imagínate que deseemos escribir en pantalla la frase: «Lo bien hecho, bien parece».
La instrucción para visualizarla en pantalla encerrada entre comillas será:

```
PRINT CHR$(34); "LO BIEN HECHO, BIEN PARECE"; CHR$(34)
```



EJERCICIOS

Empleando la función TAB, busca un nuevo método para imprimir la tabla de multiplicar de un número, de forma que las unidades, las decenas, etc., de los distintos productos queden perfectamente encolumnadas. Para ayudarte, he aquí un ejemplo con la tabla del 7.

```
10 LET C = 10: LET S = C: REM LAS UNIDADES SE IMPRIMEN EN LA COLUMNA 10
20 CLS
30 FOR V = 1 TO 10
40 LET P = V * 7
50 IFP> 9 THEN S = C - 1
60 PRINT TAB S; P
70 NEXT V
```

Sustituye ahora la línea 30 con: 30 FOR V = 1 TO 20.
Sugerencia: para encolumnar las centenas, introduce entre la línea 50 y la 60 un nuevo control.

Juega a los dados con tu Spectrum

```
 10 CLS
 20 INPUT "JUGADOR 1 ="; J$
 30 INPUT "JUGADOR 2 ="; H$
 40 PRINT AT 1,10; "DADOLOCO" ' ' ' '
 50 PRINT J$; "PULSA UNA TECLA"
 60 PAUSE 0
 70 LET R1 = INT (6 * RND) + 1
 80 PRINT AT 10; R1; ' '
 90 PRINT H$; "PULSA UNA TECLA"
100 PAUSE 0
110 LET R2 = INT (6 * RND) + 1
120 PRINT TAB 10; R2
130 IF R1 = R2 THEN PRINT ' '; TAB 4; "EMPATE": GO TO 200
140 IF R1 > R2 THEN PRINT ' '; TAB 4; "GANA"; J$: GO TO 200
150 PRINT ''; TAB 4; "GANA "; H$
200 PRINT ''; "¿SEGUIMOS? S/N"
210 PAUSE 0: LET T$ = INKEY$
220 IF T$ = "S" OR T$ = "s" THEN CLS: GO TO 40
230 CLS: PRINT AT 12,14; "FIN"
```

Modifica las líneas 70 y 110 para simular el lanzamiento de 2 dados: piensa cuales son los números máximo y minimo que pueden salir.



El fruto de la Investigación

La nueva impresora de **SEIKOSHA SP-800**, con un ordenador personal puede escribir ***96 combinaciones de letra diferentes,*** desde 96 caracteres por segundo a 20 con muy alta calidad de letra, además es gráfica en alta densidad.

Su precio es de 69.900 Pt con ***introductor automático hoja a hoja.***

Con un pequeño ordenador personal, un procesador de textos puede costar alrededor de cien mil pesetas.

Infórmese y comprenderá por qué **las máquinas de escribir** tienen demasiados años.

Nuestra calidad es "SEIKO";
nuestros precios, únicos
Si desea más información,
consulte con nuestro distribuidor
más cercano, llame o escriba a:



DIRECCION COMERCIAL:
Av. Blasco Ibañez, 114-116
46022 VALENCIA
Tel (96) 372 88 89
Télex 62220
DIRECCION COMERCIAL EN CATALURA:
C/Muntaner, 60-2-4Pta
08011 BARCELONA
Tel (93) 323 32 19

ESTOS SON NUESTROS MODELOS:

MODELO	VELOCIDAD	COLUMNAS	TIPOS DE LETRA	P.V.P.R. INTERFACE PARALELO
GP-50S LA DEL SPECTRUM	40 cps	32	-	19.900
GP-50 LA PEQUEÑA	40 cps	46	2	25.900
GP-500 LA ECONOMICA	50 cps	80	2	47.900
GP-700 LA DE COLOR	50 cps	80-106	3	69.900
SP-800 LA PERFECCION	96 cps	80-137	20	69.900
BP-5200 LA DE OFICINA	200 cps	136-272	18	199.900
BP-5420 LA MAS RAPIDA	420 cps	136-272	18	299.900

* Los precios indicados son los recomendados para conexion tipo paralelo Centronics, para otro tipo de conexion, sufren un ligero incremento.

Este pie de página ha sido realizado integramente con la nueva impresora:

SEIKOSHA SP-800